Remarks/Arguments

Claims 1-5 and 7-25 remain in this application.

The examiner has objected to the drawings under 37 CFR 1.83(a).

The examiner has objected to claims 5-8, 12, 19 and 20 under 37 CFR 1.75(c), as being of improper dependent form for failing to further limit the subject matter of a previous claim.

The examiner has rejected claims 13 and 23-25 under 35 USC 112, first paragraph, as based on a disclosure which is not enabling.

The examiner has rejected claims 5-8 and 13-25 under 35 USC 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

The examiner has rejected claims 1-17, 19-23 and 25 under 35 USC 103(a) as being unpatentable over the admitted prior art and Daniel Kovacs' "Tutorial on Linked Lists."

The examiner has objected to claims 18 and 24 as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form to overcome the rejection(s) under 35 USC 112 and to include all of the limitations of the base claim and any intervening claims.

In view of the above amendments and these remarks, reconsideration of the above noted rejections and objections is respectfully requested.

Drawing Objections:

Applicant respectfully traverses the objection to the drawings. Applicant respectfully submits that the features of claim 18 (as amended) and claim 24 are shown in the drawings.

An example of the features in claim 18 (as amended herein) is described in the paragraph beginning at page 19, line 24. At 54 in **Fig. 3**, the completion of the

remaining one delayed read command is recognized (page 19, line 30, to page 20, line 1). Then "the get queue pointer 34 is advanced [at 61 in **Fig. 3**] to the next command 32d, which in this example is the non-read command 32d, as shown in **Fig. 2R**" (page 20, lines 4-6, as amended herein). Therefore, the process flow 40 (**Fig. 3**) will apply the command 32d from the command buffer 30 to the data mover 16 for transfer to the target device 22, 23 or 24 for completion according to 50, 52 and 54 in **Fig. 3**. Additionally, the command 32d was not included in the loop before the loop was abolished at 80 in **Fig. 3** (page 19, lines 24-26) because the command 32d was not a read command, as determined at 64 in **Fig. 3** and as indicated in **Figs. 2F-2R** (see page 16, lines 10-16). Applicant respectfully submits, therefore, that the features of claim 18 (as amended) are shown in the drawings in **Figs. 2F-2R and 3**.

An example of the features in claim 24 is described beginning at page 15, line 6 (as amended herein). At page 15, lines 6-9 (as amended herein), the first delay in completing the preceding read command 32b is recognized at 54 in **Fig. 3**. Following the subsequent execution of the process flow 40 through 56, 62, 64 and 66 (page 15, lines 9-20, and **Figs. 2D, 2E and 3**), the loop end pointer 38 is advanced to the next subsequent read command 32c in the command buffer 30 at 66 (page 15, lines 18-20, and **Figs. 2E and 3**). Applicant respectfully submits, therefore, that the features of claim 24 are shown in the drawings in **Figs. 2D, 2E and 3**.

Consequently, Applicant respectfully requests withdrawal of the objection to the drawings under 37 CFR 1.83(a), since the limitations of claim 18 (as amended) are shown in Figs. 2F-2R and 3 and the limitations of claim 24 are shown in Figs. 2D, 2E and 3.

Claim Objections:

Applicant respectfully traverses the objection to **claim 5** under 37 CFR 1.75(c), as being of improper dependent form for failing to further limit the subject matter of a previous claim. Claim 5 depends from claim 2, which depends from

claim 1. Claim 5 introduces a queue pointer by which commands are selected for completion and recites:

> adjusting the position of a queue pointer to move through the commands at and between the loop start pointer and the loop end pointer until all of the commands at and between the loop start pointer and the loop end pointer have been completed.

Applicant respectfully submits that neither claim 1 nor claim 2 includes this limitation or the queue pointer.

The Office Action states that claim 1's means for scanning through the loop and selecting a command in the loop for processing is the queue pointer. However, claim 1 simply recites "attempting to complete the read commands at and between the loop start pointer and the loop end pointer," which may be done by any appropriate means. Since claim 1 merely calls for "attempting to complete," and since this portion of the method of claim 1 may be performed by any appropriate means, Applicant respectfully submits that it is not necessary to conclude that the "attempting to complete" portion of the method includes the scanning and selecting and the means for performing the scanning and selecting. Without such necessity, it is improper to read such limitations into such a claim. It is also improper then to go further and to equate such unnecessary means with an element (the queue pointer) that is in fact specifically not referred to in the claim. Proper dependent claim language, on the other hand, allows the introduction of a limitation in a dependent claim that was not in fact in a previous claim.

Consequently, Applicant respectfully requests withdrawal of the objection to **claim 5** under 37 CFR 1.75(c), since claim 1 does not provide the limitations of claim 5.

Applicant respectfully traverses the objection to **claims 6-8** under 37 CFR 1.75(c), as being of improper dependent form for failing to further limit the subject matter of a previous claim. Claim 6 has been cancelled by the above amendments. Claim 7 previously depended from claim 6, but has been amended to depend from claim 5. Claim 8 depends from claim 7. Applicant respectfully submits, therefore,

that the grounds for the objection to claims 6-8 have been cured. Consequently, Applicant respectfully requests withdrawal of the objection to **claims 6-8** under 37 CFR 1.75(c).

Applicant respectfully traverses the objection to **claim 12** under 37 CFR 1.75(c), as being of improper dependent form for failing to further limit the subject matter of a previous claim. Claim 12 depends from claim 1. Claim 12 has been amended by the above amendments. Amended claim 12 recites:

> for each subsequent read command subsequent to the aforementioned first, second and third read commands, advancing the loop end pointer to the next subsequent read command in the queue upon experiencing a delay in completing the read command previously identified by the loop end pointer.

Claim 12 has, thus, been amended to specifically refer to read commands subsequent to the read commands referred to in claim 1. In other words, the limitation of amended claim 12 expressly does not repeat claim 1's 5th and 7th limitations, since the limitations refer to different read commands. Consequently, Applicant respectfully requests withdrawal of the objection to **claim 12** under 37 CFR 1.75(c), since claim 12 does not merely repeat limitations of claim 1.

Applicant respectfully traverses the objection to **claim 19** under 37 CFR 1.75(c), as being of improper dependent form for failing to further limit the subject matter of a previous claim. Claim 19 previously depended from claim 17, which depends from claim 14, which depends from independent claim 13. However, claim 19 has been amended by the above amendments to depend from claim 14. Amended claim 19 recites:

> ... programmed logic functionality which:
> applies the delayed read commands of the loop to the data mover for transfer to the target device for completion until the loop of delayed read commands is abolished ....

The phrase "until the loop of delayed read commands is abolished" is a limitation that is not in either claim 13 or 14. Consequently, Applicant respectfully requests

withdrawal of the objection to **claim 19** under 37 CFR 1.75(c), since claim 19 includes a limitation not in preceding claims 13 and 14.

Applicant respectfully traverses the objection to **claim 20** under 37 CFR 1.75(c), as being of improper dependent form for failing to further limit the subject matter of a previous claim. Claim 20 depends from claim 14, which depends from independent claim 13. Claim 20 introduces a queue pointer and a **normal** "queue pointer positioning functionality of the interface controller." Claim 20 then further introduces:

> ... programmed logic functionality which:
> **overrides the normal** positioning of the queue pointer positioning functionality of the interface controller to control the positioning of the queue pointer when the loop has been established.

Applicant respectfully submits that there is no reference in either claim 13 or 14 to a queue pointer, normal queue pointer positioning functionality or an ability to override the normal queue pointer positioning functionality.

The Office Action states that claim 20's limitation is claim 14's last limitation. However, the last limitation of claim 14 calls for:

> ... programmed logic functionality which:
> ...
> applies the delayed read commands of the loop to the data mover for transfer to the target device for completion before applying any other read commands from the command buffer to the data mover for transfer to the target device.

Applicant respectfully submits that there is no reference in this limitation to a "queue pointer" or to a "queue pointer positioning functionality," so there can be no implication or inference of a "**normal** positioning of the queue pointer positioning functionality." Furthermore, without the "normal positioning of the queue pointer positioning functionality," there can be no implication or inference of an **override** of the "normal positioning of the queue pointer positioning functionality."

Consequently, Applicant respectfully requests withdrawal of the objection to **claim 20** under 37 CFR 1.75(c), since claims 13 and 14 do not provide the limitations of a queue pointer, a normal positioning of a queue pointer positioning

functionality or an override of the normal positioning of the queue pointer positioning functionality.

Rejections under 35 USC 112:

Applicant respectfully traverses the rejection of **claims 13 and 23-25** under 35 USC 112, first paragraph, as based on a disclosure which is not enabling. Claim 13 is independent, and claims 23-25 depend from independent claim 13. The Office Action states that lines 26-28 of page 4 of the application provides that limiting the queue pointer within the loop is critical or essential to the practice of the invention, but that this feature is not included in these claims. Applicant respectfully submits, however, that page 4, lines 26-28 merely states that a method to which the present invention **"relates"** comprises in part:

> attempting to complete the read commands at and between the loop start pointer and the loop end pointer until all of those read commands have been completed, before attempting to complete other commands in the queue.

This quoted feature is identical to the last limitation in claim 1. Applicant respectfully submits, however, that there can be no statement or inference that a feature is critical or essential to the practice of an invention when the feature is part of a method to which the present invention merely "relates." Thus, the quoted feature is, itself, not stated as being critical or essential to the practice of the invention. Furthermore, there is no reference to a "queue pointer" in the quoted feature, so there also can be no requirement of "limiting the queue pointer." Applicant respectfully submits, therefore, that it is improper to find a critical or essential limitation (limiting the queue pointer as mentioned) requiring an element (the queue pointer) that is neither expressed nor implied in a feature (the quote above) that is only "related" to the present invention.

Additionally, Applicant respectfully submits that the relevant reference to the queue pointer is to be found at page 6, lines 1-22, wherein the application states that "**Other aspects** of the queue processor of the interface controller include programmed logic functionality which" (page 6, lines 1-2), among other functions, "limits the position of the queue pointer to those commands identified by and

between the loop start pointer and the loop end pointer" (page 6, lines 20-22). The interface controller, again, merely "relates" to "another specific aspect" of the present invention. (page 5, lines 8-9) Thus, the limiting of the position of the queue pointer is merely another functional aspect (not critical or essential) of an element (the queue processor) of a device (the interface controller) that is merely another specific aspect (another embodiment) of the present invention.

Consequently, Applicant respectfully requests reconsideration and withdrawal of the rejection of **claims 13 and 23-25** under 35 USC 112, first paragraph, since the quoted feature does not describe a limitation not included in these claims that is critical or essential to the practice of the invention.

Applicant respectfully traverses the rejection of **claims 5-8 and 13-25** under 35 USC 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claim 5 has been amended by the above amendments to change the reference to "a queue pointer" previously in line 3 (now in line 4, above amendment) to "the queue pointer," which has its proper antecedent basis in the reference to "a queue pointer" in line 2. Additionally, **claim 6** has been cancelled by the above amendments, and **Claims 7 and 8** now depend directly or indirectly from claim 5. Consequently, Applicant respectfully requests reconsideration and withdrawal of the rejection of **claims 5-8** under 35 USC 112, second paragraph, in light of the amendment to claim 5.

Claim 13 has been amended by the above amendments to change the reference to "a target device" in line 10 to "the target device," which has its proper antecedent basis in the reference to a "target device" in line 2. Additionally, **claims 14-25** depend either directly or indirectly from independent claim 13. Consequently, Applicant respectfully requests reconsideration and withdrawal of the rejection of **claims 13-25** under 35 USC 112, second paragraph, in light of the amendment to claim 13.

Claim 15 has been amended by the above amendments to change the

reference to "a read command" in line 4 to "each read command," which has its proper antecedent basis in the references to "read command" in claims 13 and 14, from which claim 15 depends. Consequently, Applicant respectfully requests reconsideration and withdrawal of the rejection of **claims 15** under 35 USC 112, second paragraph, in light of the amendments to claims 13 (previous paragraph) and 15.

Rejections under 35 USC 103(a):

Claim 1:

Applicant respectfully traverses the rejection of claim 1 under 35 USC 103(a) as being unpatentable over the discussion of *head-of-list alternation* in the application and *Daniel Kovacs'* "Tutorial on Linked Lists."

Claim 1 calls for (previously at lines 13-19; now at lines 13-18, above amendments):

> determining that the command in the queue following the second delayed read command is a third read command;
>
> advancing the loop end pointer to identify the third read command; and
>
> attempting to complete the read commands at and between the loop start pointer and the loop end pointer until all of those read commands have been completed, before attempting to complete other commands in the queue.

Applicant respectfully submits that *head-of-list alternation* and *Kovacs* do not teach or fairly suggest, either singularly or in combination, these limitations.

Head-of-list alternation is described in the application from page 2 at line 15 to page 4 at line 9, as amended above. Briefly stated, *head-of-list alternation* works when there are two sequential read commands at the head of the list of commands in the queue and at least the first one is delayed. When the command at the head of the list is determined to be a delayed read command and the next command is another read command, then *head-of-list alternation* effectively **swaps the position** of the two commands to put the second read command at the head of the list. Then an attempt can be made to execute and complete the second read command as if it were the first read command. However, if completion of the second read command

is also delayed, then the same procedure is used to effectively swap the two commands again. In other words, since the command at the head of the list (i.e. the original second command) is determined to be a delayed read command and since the next command is another read command (i.e. the original first command), the two commands can be swapped back to their original relative positions. In this manner, the command at the head of the list **alternates** (hence the name "*head-of-list alternation*") between the top two read commands until one of them completes.

After each alternation, the command at the head of the list is rechecked to determine if it is still delayed or has completed. This procedure can continue as long as the two read commands are delayed. In this manner, *head-of-list alternation* always attempts to operate on the read command at the head of the list, but if it is delayed, then it is swapped with the next read command.

After one of the two read commands at the head of the list completes, the completed command is removed from the queue, and whichever command remains is the first command in the queue. If this command is still delayed, then the procedure checks the next command (now the new second command in the queue) to determine whether it too is a read command. If so, then the top two commands can again be alternated until one of them completes. In other words, *head-of-list alternation* can work on two **and only two** read commands at a time. *Head-of-list alternation* has not been made to work on three commands at the same time because "substantial complexities are encountered when attempting to expand the number of delayed read commands beyond two, particularly in regard to handling those delayed read commands that may have been completed between the first and the last ones of a greater number of delayed read commands" (page 3, lines 26-30).

The Office Action states that it involves only routine skill in the computer art to duplicate the "recognizing the second read command" in a *head-of-list alternation* procedure to provide a "recognizing the third read command" element for claim 1. However, as explained above, *head-of-list alternation* works only for two read commands, not three or more. Therefore, it cannot be a matter of routine skill in the

computer art to extend *head-of-list alternation* to three read commands. Instead, since *head-of-list alternation* always operates on the read command at the head of the list, every time the procedure "recognizes the **next** read command," the procedure is actually "recognizing the **second** read command." Therefore, the procedure already does duplicate the "recognizing the second read command," as stated in the Office Action, but each time it is always the **second** read command that is recognized. As a consequence, *head-of-list alternation* will not "determin[e] that the command in the queue following the second delayed read command is a third read command" (an element of claim 1, quoted above), since *head-of-list alternation* will not even look at the third command until it becomes the second command, in which case the method of claim 1 is not being followed at all.

Additionally, since *head-of-list alternation* cannot be extended to work with three or more read commands, *head-of-list alternation* also cannot be adapted to include the features of linked lists, as described in *Kovacs*. Thus, *head-of-list alternation* cannot "advanc[e] [a] loop end pointer" (another element of claim 1, quoted above). Furthermore, since *head-of-list alternation* can neither work with more than two read commands nor be adapted according to *Kovacs* to include the features of linked lists, *head-of-list alternation* cannot attempt "to complete the read commands at and between the loop start pointer and the loop end pointer until all of those read commands have been completed" (yet another element of claim 1, quoted above). Instead, *head-of-list alternation* can only attempt to complete the read command that is currently at the head of the list, and linked list features cannot change this capability.

Applicant respectfully submits, therefore, that independent claim 1 is not anticipated by, is not obvious from, and is patentable over the discussion of *head-of-list alternation* in the application and *Daniel Kovacs'* "Tutorial on Linked Lists," since the combination thereof does not teach or fairly suggest the quoted limitations.

Claims 2-5 and 7-12:

Applicant respectfully traverses the rejection of claims 2-5 and 7-12 under 35 USC 103(a) as being unpatentable over the discussion of *head-of-list alternation* in the application and *Daniel Kovacs'* "Tutorial on Linked Lists." Claims 2-5 and 7-12 depend directly or indirectly from claim 1. As explained above, claim 1 is not anticipated by, is not obvious from, and is patentable over *head-of-list alternation* and *Kovacs*. Therefore, Applicant respectfully submits that claims 2-5 and 7-12 are not anticipated by, are not obvious from, and are patentable over *head-of-list alternation* and *Kovacs* for the same arguments regarding claim 1, above.

Claim 13:

Applicant respectfully traverses the rejection of claim 13 under 35 USC 103(a) as being unpatentable over the discussion of *head-of-list alternation* in the application and *Daniel Kovacs'* "Tutorial on Linked Lists."

Claim 13 calls for (previously at lines 24-29; now at lines 24-28, above amendments):

> determines that the command in the command buffer following the second read command is a third read command;
> advances the loop end pointer to identify the third read command; and
> applies the third read command from the command buffer to the data mover for transfer to a target device for completion.

Applicant respectfully submits that *head-of-list alternation* and *Kovacs* do not teach or fairly suggest, either singularly or in combination, these limitations.

The explanation of *head-of-list alternation* is given above with respect to the discussion of claim 1. Additionally, the first two quoted elements of claim 13 (previous paragraph) are similar, though not identical, to the first two quoted elements of claim 1. Therefore, it follows that *head-of-list alternation* will not "[determine] that the command in the command buffer following the second read command is a third read command" (first quoted element of claim 13) for the same reasons outlined in the discussion of claim 1, i.e. since *head-of-list alternation* will not even look at the third command until it becomes the second command.

Additionally, it also follows that *head-of-list alternation* cannot "[advance] the loop end pointer to identify the third read command" (second quoted element of claim 13) for the same reasons outlined in the discussion of claim 1, i.e. since *head-of-list alternation* cannot be adapted to include the features of linked lists. Furthermore, since *head-of-list alternation* will not even look at the third command until it becomes the second command, *head-of-list alternation* will not "[apply] the third read command from the command buffer to the data mover for transfer to a target device for completion" (third quoted element of claim 13).

Applicant respectfully submits, therefore, that independent claim 13 is not anticipated by, is not obvious from, and is patentable over the discussion of *head-of-list alternation* in the application and *Daniel Kovacs'* "Tutorial on Linked Lists," since the combination thereof does not teach or fairly suggest the quoted limitations.

Claims 14-17, 19-23 and 25:

Applicant respectfully traverses the rejection of claims 14-17, 19-23 and 25 under 35 USC 103(a) as being unpatentable over the discussion of *head-of-list alternation* in the application and *Daniel Kovacs'* "Tutorial on Linked Lists." Claims 14-17, 19-23 and 25 depend directly or indirectly from claim 13. As explained above, claim 13 is not anticipated by, is not obvious from, and is patentable over *head-of-list alternation* and *Kovacs*. Therefore, Applicant respectfully submits that claims 14-17, 19-23 and 25 are not anticipated by, are not obvious from, and are patentable over *head-of-list alternation* and *Kovacs* for the same arguments regarding claim 13, above.

Allowable Subject Matter:

Applicant thanks the Examiner for indicating that claims 18 and 24 would be allowable if rewritten in independent form. Claims 18 and 24 depend from independent claim 13. Due to the above arguments regarding independent claim 13, above, claims 18 and 24 have not been rewritten in independent form. Rather, Applicant respectfully requests withdrawal of the objection to claims 18 and 24 as

being dependent upon a rejected base claim, since claim 13 is not anticipated by, is not obvious from, and is patentable over *head-of-list alternation* and *Kovacs*.

Additional Amendments to the Specification:

Additional amendments, not mentioned in the above remarks, have been made to the Specification to correct minor typographical errors discovered upon reviewing the Application.

For the reasons specifically discussed above, and others, it is believed that pending claims 1-5 and 7-25 define patentable subject matter. Reconsideration of the previous rejections as they might apply to the pending claims is therefore respectfully requested. Applicant respectfully requests that a timely Notice of Allowance be issued in this case.

Respectfully submitted,

September 15, 2004
Date

L. Jon Lindsay
Registration No. 36,855
ATTORNEY FOR APPLICANT

L. Jon Lindsay
1857 W. Spring Water PL
Highlands Ranch, CO 80129
Telephone: 720-344-6189
Facsimile: 928-563-4114